Investigating Odometry error caused during collisions due to bump sensors

2073015 2071959

***Abstract*—Collisions are generally undesired in mobile robots. Most of the systems use a submodule of collision avoidance. In this report, we investigate the errors caused due to collision and how it varies according to different robot parameters, i.e. speed and angles. In the 3pi+ robot, we made an assumption based on existing literature on how it creates an Odometry inaccuracy. Our results use the mean and variance of the recorded error to explain the error magnitude in different scenarios.**

I. Introduction

The 3pi+ Pololu robot comes with encoder-based kinematics. Here we look into the background of our robot and its motivation to do our experiment.

A. Background

Mobile robots' positioning system is an integral part of their operation worldwide. It is responsible for the robot to move to a specified location or just general operational tasks. An error in this may result in drastic consequences in real-world scenarios.

In this paper, we are working with a Pololu 3pi+ robot. According to the robot's system modules from the website, localization in the world is based on processing the encoder values attached to the robot's motors to get the kinematics for the robot. This is a sound system as it is based on real-world feedback. However, the encoder-based kinematics used here also have limitations. For Example, when our robot collides with an object in operation, it will cause an odometry error which will, in turn, affect the kinematics of the robot. This may be due to slipping or jumping when the robot comes into contact with the object. This also increases with time as the robot moves in the environment.

B. Motivation

We now understand the impact of collisions on the kinematics of the 3pi+ robot. This report aims to investigate the impact of collisions on the kinematics of the robot in different experimental setups.

During our literature review on the previous studies done with collision influence on robots, we understood that they play a significant part in the smooth working of the robot system. We understood that in some cases where the robot has other systems to understand the environment, collision helps give more information about the environment allowing it to correct itself and perform ideally [1]. In some cases, the robot is made to understand the probability of future collisions and avoid them altogether. There are more workflows in case of collision events, but generalized ones are these [2].

C. Hypothesis Statement

Wheel slippage results from collisions with environmental impediments. It affects the odometry because our odometry system depends on the rotation of the wheels directly correlating to the robot's movement. Our hypothesis is that:

> By combining the output from the two bump sensors with the kinematics data, we determine how much wheel slippage there is when the Pololu 3pi+ collides with stationary objects. We speculate that the error increases with an increase in the robot's speed and collision angles.

We investigate this hypothesis using a structured experiment using the mobile Pololu 3Pi+ robot, correcting any underlying inaccuracies to isolate the inaccuracy brought on by bump sensors in odometry.

II. Implementation

For the robot to even undertake an experiment to demonstrate the error caused by collisions, it must be aware of its place in the world. To achieve this, we have employed a technique that computes the rotational difference between encoder counts over time [3], [4]. We then translate these values to a real-world observation standard of speed and distance from the predetermined robot dimensions, such as the wheel radius, count per encoder rotation, gear ratio of the motor, and the distance the robot can go in a single wheel spin. Usually, vehicles taking turns use an arc as their path, but this approach considers the robot rotating about its axis to reduce error generation. We utilise a PID controller to keep the robot moving at a constant speed, and in some instances of the experiment, we also employ a heading controller that tracks heading direction using encoder difference to ensure the wheels do not deviate from the course. These enable the robot to travel straight ahead, which is essential for our routine. As we state in our hypothesis, an error can be developed in the kinematics due to bump sensors that detect a collision of sorts.

A. System Structure

The routine is that our robot goes in a straight line, collides, turns around, and reaches what it considers the home or origin of the global frame of reference in that iteration. There is a difference between what it considers home to its physical world location of the home. This is due to the odometry added up due to the collision. Now there was a need for us to decide the robot's behaviour after its collision. We identified two scenarios; One is the robot goes reverse, and the other is it rotates to the home location, though here there was a case of the extra unrelated error caused by the contact forces by the object during the turn. However, after colliding with the object, we observed that the robot is not in contact with it, so when it turns, the contact forces do not influence it. So we adopted the turnaround behaviour for the robot. We did different investigations with various environment settings to investigate the behaviour such that countermeasures that are non-expensive but rather the matter of change of behaviour of the robot in response is encouraged. This will also apply when scaled to behaviour in more prominent and sophisticated robots in the real world.

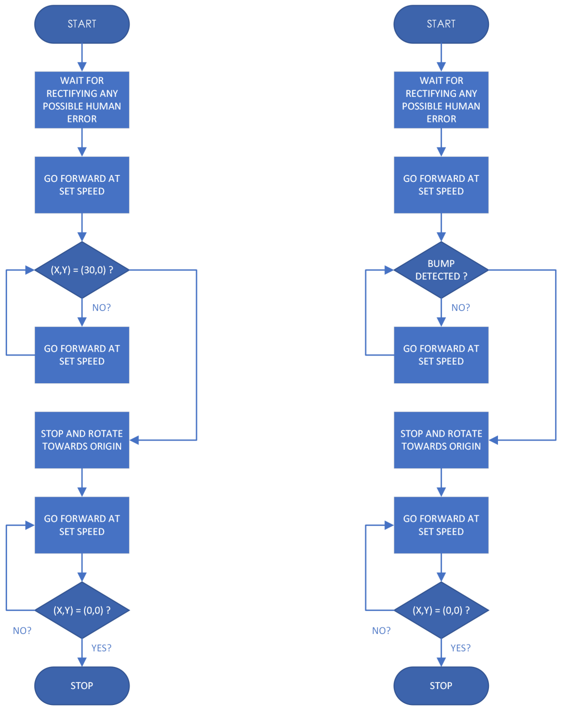


Fig. 1: Flowchart showcasing our code structure for the experiment

B. Bump Sensor

Bump sensors on the 3pi+ robot help us detect our collisions. With a few minor operational differences, bump sensors operate similarly to line sensors. However, instead of IR light reflected off the ground, it is measured off the two flexible plastic flaps or segments (bumpers) at the front of the 3Pi+ robot [3]. When the plastic flaps are pressed against the 3Pi+ body, the time it takes for light to reflect reduces. This can further be used to detect any obstacle or the kind of ambient light condition it is present.

We observed the trend for the readings when the flaps were compressed 30 times and then calculated the mean of those values to calibrate the bump sensor data to reflect what is regarded as a bump and can be further quantified to represent the amount of force with which the collision happened. Calibration should be conducted in a setting we regard as an experimental setup since the sensors are sensitive to the local ambient light levels. We can also make the robot move in a pre-defined obstacle-filled environment, using this particular set of sensors to map out the obstacles.

$$Threshold = 1.1 * \frac{1}{30}\sum_{i=1}^{30} a_i \tag{1}$$

C. Kinematics

A general kinematics model using the encoders for each of the motors present on the 3pi+ is used for calculating the robot's position in a global coordinate frame or the experiment environment, considering the origin as its starting point. Robot specifications such as wheel diameter, the distance between the wheels, the distance covered per rotation of the encoder, and the distance covered per rotation of the wheel are required to determine the robot's pose and orientation. The local coordinates of the robot, such as X-contribution *x(r)*, Y-contribution *y(r)* and Orientation *r(r)* are calculated from the left and right encoder distances θ_l and θ_r using the following equations (*y(r)* is '0' because there is no translation possible horizontally considering the robot architecture),

$$x(r) = \frac{r\theta_l}{2} + \frac{r\theta_r}{2} \tag{2}$$

$$r(r) = \frac{r\theta_l}{2l} - \frac{r\theta_r}{2l} \tag{3}$$

In equation 3, 2l is the distance between wheels.

A mobile robot in an experiment always changes its location over time, making it easier for us to track its position and orientation in a global frame. Using equations 2 and 3, we can obtain the X, and Y coordinates *x(i)*, *y(i)* and the orientation of the robot *r(i)* in the experiment frame in equations of time. [5], [6]

$$x(i)^{t+1} = x(i)^t + (x(r)\cos(r(i)^t)) \tag{4}$$

$$y(i)^{t+1} = y(i)^t + (x(r)\sin(r(i)^t)) \tag{5}$$

$$r(i)^{t+1} = r(i)^t + r(r) \tag{6}$$

D. PID

For our robot to have smooth performance and perfect heading control, PID is inevitable. We have divided the PID control into two specific parts: speed control and heading control. Heading control has not been used in the main part of the experiment considering a real-world scenario of mapping an unknown location but was used to compare the performance. [7]

In all scenarios, our robot's motion should be free of jerks, which occur when it makes abrupt or significant changes to its motion. For our small robot, rapid changes to wheel speed are likely to cause the wheels to slip on the surface, and if this happens, there will be encoder counts, but the robot will not change its pose in reality. Any prevention of sudden movements or motions can only be prevented with the use of PID, which provides continuous input for the motors to head in a specific direction or maintain a specific speed from the rotation or a point to travel; the speed of rotation of wheels. PID controller is to read a sensor (*m(t)*) and then compute the desired actuator output by calculating proportional, integral, and derivative responses and summing those three components to compute the output (*d(t)*). Heading control is generally a feedback loop for the heading direction using encoder difference to ensure the robot faces the point where the course is charted.

The error signal can be calculated from the equation 7 below,

$$e(t) = d(t) - m(t) \tag{7}$$

We use the following equation and implement this in our system as our PID controller,

$$Output = K_P * e(t) + K_i \int_0^t e(t), dt + K_d * \frac{de(t)}{dt} \tag{8}$$

III. Experiment

A. Overview of Method

Our experiment involved a few steps: setting up the experiment place, aligning the robot, choosing an object that can act as a perfect obstacle, and collecting data after the robot completes its routine.

1) Initial setup of the experiment: To ensure we can collect our data correctly and ensure there is no human error, we used black sealing tape and drew a square. The edges' centres are marked, so the robot can be appropriately aligned; also, the centre of the box is marked. Now, a line of 30 cm has been drawn from one edge of the box. An obstacle with considerable weight is chosen for the experiment and placed at the other end of the line.
2) Now, the robot is aligned correctly for the experiment using the markers and the box, such that it is facing in the direction it should go.
3) The routine for the robot starts by giving us leeway to ensure it is aligned properly; then, it takes its initial position as the origin of the global frame and starts going forward. It will move straight until it hits an obstacle, and then the bump sensors of the robot, which are situated at the very front, get a fluctuation in values. If these cross a threshold, the robot stops its motion and records its location. The new position is marked, calculates its position from the origin, and tries to return. The error in its interpretation of origin from the physical world is calculated.
4) This is done ten times for every experiment, considering different external conditions.

B. Baseline Experiments

Here, before we go into the odometry error produced by collision, we need to eliminate the inherent odometry errors of the robot kinematics, which generally cause the robot to perceive its location differently in its calculations compared to its physical location on the plane. If we are to use this robot directly to calculate any error, these inherent errors may either increase or decrease the error caused just by the experimental method. So, we eliminated those kinematics errors by removing the mean bias from the values collected from the following tests and deciding which to adopt for collecting the controlled bump sensor error. The Line following method was used to make the robot move in a straight line of 30 cm and return to its initial position. This was repeated ten times, and the mean of the errors generated is stored; we can see this in *(fig.2)*.

Now the robot was moved in a straight line without line following, but with heading control for the direction with PID speed control to maintain its speed and direction of heading and the mean error generated is stored. The stored value from the line following and then with heading control are compared to get a reference of values we should get in an optimal condition while moving in a straight line. This helped us rectify our PI controller and heading control for errors and get results closest to the optimal values.

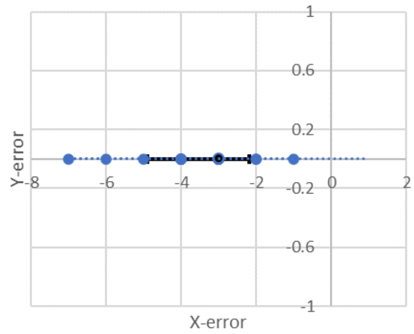
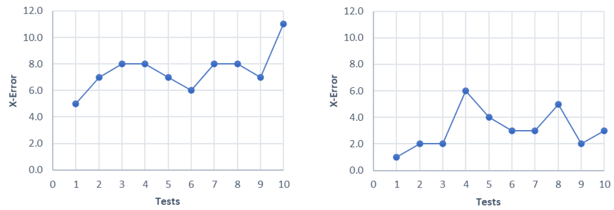


Fig. 2: Error with line following



(a) Origin to Imperative location, error in x (mm)

(b) Imperative to Origin, error in x(mm)

Fig. 3: Baseline experiment graphs

Finally, the robot was operated through speed control PID without heading control, as in a real-life situation, we do not always have a destination, making the heading control redundant. The mean error obtained from the experiment *(fig.3)* that only involves PID speed control is used for testing the required bias for inherent kinematics error.

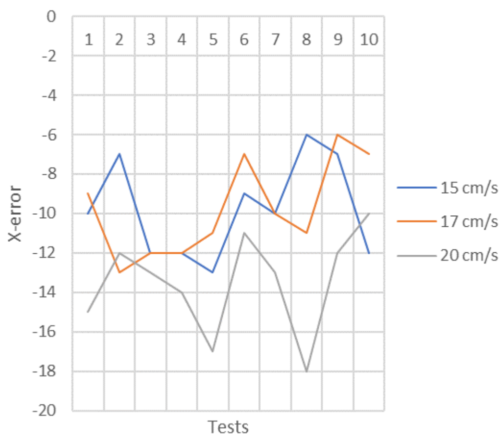


Fig. 4: Errors generated in X(mm) for speeds 15cm/s, 17.5cm/s and 20cm/s respectively

Now, we used different speeds, 15 cm/s, 17 cm/s and 20 cm/s and made the robot complete its routine ten times on each speed and validated the obtained mean error as the inherent bias for this particular robot as the perceived origin location from robot calculations and physical location of the global frame are the same *(fig.4)*.

We are considering the bias from the X and Y position of the robot here. However, after conducting several experiments, we now will not be considering the Y coordinates for investigation of the robot error further in the experiment as there is a right-oriented bias in the robot causing an unstable error, which is a systematic error generated by the limiting hardware of the robot [8].

C. Experiments with different Speeds

So, in the previous experiments, the base performance of the system was quantified, and the inherent errors were mitigated. Now we evaluate the collision errors caused by different conditions of speeds. These errors are correlated with the speeds to find the effect caused by collision and conclude about the optimal speed for our further experiments. First, we conducted the routine at a speed that our robot was used to in some of our previous experiments.

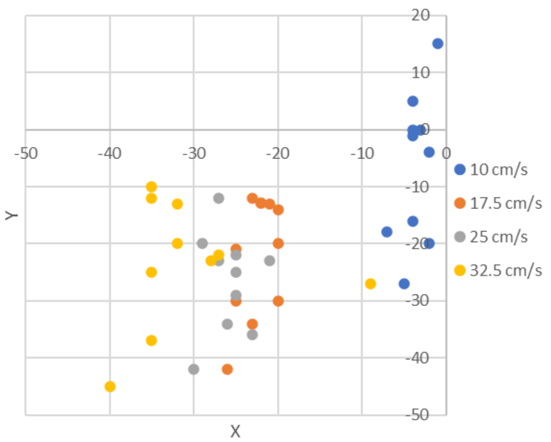


Fig. 5: Errors generated(mm) in speeds 10cm/s, 17.5cm/s, 25cm/s and 32.5cm/s respectively

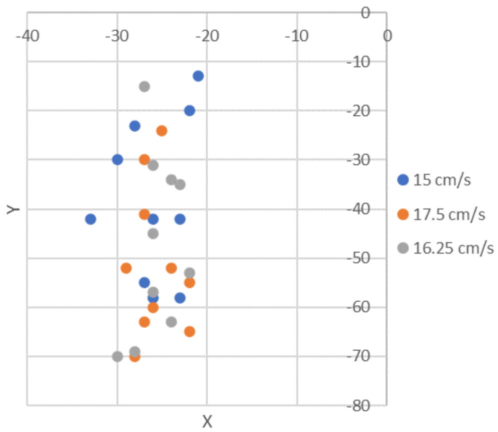


Fig. 6: Errors generated(mm) in speeds 15 cm/s, 17.5 cm/s and 16.25 cm/s respectively

A constant increment in speed is considered, and the experiment is conducted at three more speeds. So we have conducted our experiment at four different speeds 10 cm/s, 17.5 cm/s, 25 cm/s and 32.5 cm/s *(fig.5)*.

However, we have taken increments of a significant difference to find the threshold speed. So, we conducted the experiment routine with speeds near 17.5 cm/s as the error generated at this speed was significantly low, and it also completed the routine in a reasonable amount of time. The speeds chosen are 15 cm/s, 16.25 cm/s and 17.5 cm/s *(fig.6)*. The ideal speed was 16.25 cm/s; this is explained in our Results section.

A speed of 16.25 cm/s is used to obtain data with collisions at various angles in the next part of the investigation.

D. Experiments with different Collision Angles

In the experiment till now, the obstacle was placed perpendicular to the direction of the robot's heading. Considering that angle as zero, we now rotate the obstacle about the direction of heading to a certain angle, and it is considered positive for clockwise rotation and negative for anti-clockwise rotation. We are considering a small angle of 15 degrees and a large angle of 35 degrees for both clockwise and anti-clockwise directions.

The routine for these angled obstacle experiments would be the same, starting with a small quantity of delay to rectify any human errors, going forward with the help of a PID speed controller for about 30 cm and hitting the now rotated obstacle at a certain angle, stop for a small amount of time to calculates its position, store it, calculate its heading direction and head home (global frame origin).

We started the experiment from the large angle in the anti-clockwise direction, i.e., -35 degree angle and proceeded to increase the angle for every ten iterations and noted the errors as we proceeded (fig.7). If we are to rotate the object in the anti-clockwise

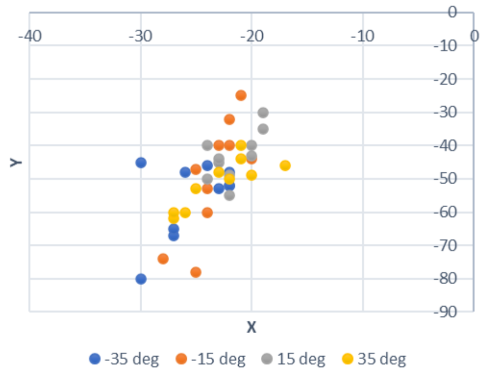


Fig. 7: Errors generated (mm) at angles -35 degrees, -15 degrees, 15 degrees and 35 degrees respectively

direction, the object is hit by only the left flap of the robot, and in the clockwise direction, it is hit by the right flap of the robot as they are perfectly divided in between and even a slight angle change from ideally zero would change the flap with which the robot hits an obstacle.

E. Discussion of Variables

What follows is a list of variables we controlled to improve the reliability and reproducibility of results:

1) Controlled Variables:

- Battery: During our previous work with Pololu 3pi+, we found that the battery plays a very significant role in the performance of the robot; the amount of charge remaining in the battery dramatically affects the odometry and if we are trying to calibrate the bot for a route, using new battery every time is the best choice. So for every 50 runs of the robot, a new recharged set of batteries were used.
- Ambient Light: The surrounding ambient light was controlled. We experimented in a room with the curtains done and the same lights on. This was done not to affect the pretty sensitive bump sensors, so we might need to calibrate it every time we change the experiment setup.
- Distance of travel: The travel distance from the start point to the collision or return point was controlled by measuring it with tape. As we are measuring the error in a robot's odometry, distance plays a significant value, and a minute change can cause all the data collected from that point useless for the current setup.
- The initial condition: The start position and angle of the robot was controlled with the making of a square encompassing the robot's dimensions, and the orientation was kept in check with the marks made in the centre of each edge of the square. Considering the global coordinate system of the robot and its physical location, the start position should be appropriately recorded.*(fig.11)*
- Angle of collision: For this, a standard scale for measurement was used, and different angles were marked during the respective experiment. Angled collisions can cause unplanned rotation of the robot, which would cause errors in calculating its position.
- Surface friction was kept using the same floor for all the experiments and regularly cleaned to reduce variation due to dust and dirt. There is always a specific difference in the speed our PID controller calculates and gives and the natural world speed of the robot due to friction. Using the same setup for the complete experiment will help remove human error.

2) Independent Variable(s): Our study is related to investigating the odometry error caused by the bump sensors. We had to simulate different types of collisions. For this reason, the speed of the robot at which it collides and the angle at which the collision occurs are varied. This helped us understand the nature of collisions as we looked into the varying angles at a constant, optimal speed.

3) Dependent Variable(s): Distance was the main dependent variable here. The x and y coordinates are from the starting point of the robot in this iteration to the current point recorded after coming back from collision. This helped us understand the diversion of the robot from the actual assumed path of the robot. We have only considered the x values for our analysis, understanding that our y values are too varied to make an inference [8]. The data collected from the bump sensor can also be considered a measurable variable affected by the speed and angles.

F. Discussion of Metric(s)

Two primary metrics of comparison between data were used, mean and variance of the error.

The mean helped in showcasing the average effect of a robot for a set of iterations in an experimental setting, which would help us understand how the robot reacts to a specific metric change in the parameters, and it also makes it easier for us to compare the obtained for understanding and rectifying the intrinsic errors in the robot.

We also calculated the variance of the error to show the position's variance for the same speed and angle condition taken in the experiment. The larger the values are, the more unstable they are. The smaller the values are, the lesser the range of error distribution and generally more stable the performance.

We are taking ten samples each so that the mean will give us an average error, and the variance of the error will give us the dispersion of data, thus giving us an understanding of the system performance and thus confirming our hypothesis.

IV. Results

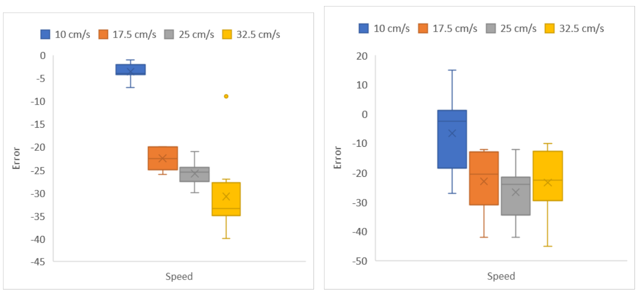
A. Baseline Results

To eliminate the initial error in Experiment section B, we have taken a couple of speeds to recreate our experiment steps without collision. The results were recorded, and the mean produced was used to eliminate the base error of our robot. Our bias is 7.5 from the start position to the robot collision point, and 3.1 was found to be in the case of the robot returning from the said collision place to the start position. This was considered a bias, and the same magnitude was removed from our code by adding it.

This removes the inherent error, so we can satisfactorily show that the errors recorded in the following change in parameters are solely due to them.

B. Results with Different Speeds (draft)

We have plotted the data collected by our experiment with various speeds, showcasing the error and variance. The variance of the error allows us to intuitively see the effect of speed and conclude its reaction to the odometry error. *(fig. 8)*

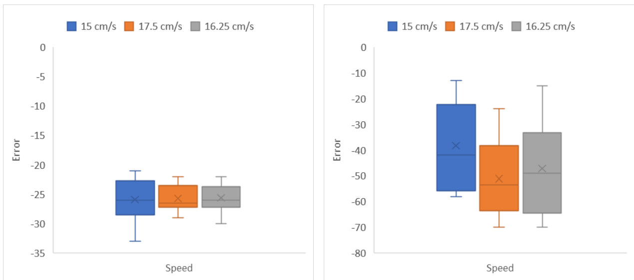


(a) X Errors(mm) generated in speeds 10 cm/s, 17.5 cm/s, 25 cm/s and 32.5 cm/s respectively

(b) Y Errors(mm) generated in speeds 10 cm/s, 17.5 cm/s, 25 cm/s and 32.5 cm/s respectively

Fig. 8: Variance of error for different speeds

Here, in these plots, we observe the variation in the error values in the experiment. As we can see, for the x coordinates of the errors measured, the least variation in error is at 10 cm/s. This speed shows significantly fewer errors. At 32.5 cm/s, we see huge errors with a relatively large deviation. However, we can see that the error significantly varies across different speeds for the Y coordinates. This again confirms our decision not to consider the y values. We can, from this data, get an estimate of how the speed at large values increases the error, and at small values is manageable. This information is used to find a threshold speed for our further experiments.



(a) X errors(mm) for speeds 15 cm/s, 17.5 cm/s and 16.25 cm/s respectively

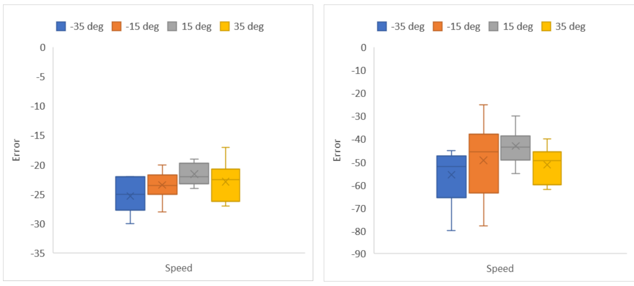
(b) Y errors(mm) for speeds 15 cm/s, 17.5 cm/s and 16.25 cm/s respectively

Fig. 9: Variance of error for different angles

In *(fig. 9)*, we can see the variation in error values for the X and Y values. For the Y values, the minor variance was seen at 16.5 cm/s, and the largest was at 15 cm/s. For the Y values, the least was 17.5 cm/s, and the most significant was 15 cm/s. Due to the minor variation in x error in 16.25 cm/s, this was chosen as our optimal speed for collision.

C. Different hitting Angle results (draft)

The trend of error has been observed and plotted here with different angles.



(a) X error(mm) for angles -35 degrees, -15 degrees, 15 degrees and 35 degrees respectively

(b) Y error(mm) for angles -35 degrees, -15 degrees, 15 degrees and 35 degrees respectively

Fig. 10: Variance of error for different angles

Now, in *(fig. 10)*, we are seeing the variance in the error values; this gives an insight into how the collision angles affect the error in our system. A consistent pattern observed from all the errors in this and previous sections error is that we are always getting the y errors more significant than the x errors. Here we can observe Now variation in error at 15 deg of collision. Whereas at other angles, the error variation is minimal and in a particular range in the x direction, the error is significant in the y direction, and the values are distributed very differently.

We can also see the angles are incremented positively, i.e. the right flap of the robot collides with the object, and y values are decreasing linearly. This shows us that the robot is indeed right-biased.

Here we have studied the effect of different collision angles and showcased how they affect our system.

V. Discussion and Conclusion

We have initially hypothesised that:

> By combining the output from the two bump sensors with the kinematics data, we determine how much wheel slippage there is when the Pololu 3pi+ collides with stationary objects. We speculate that the error increases with an increase in the robot's speed and collision angles.

Our study supported our hypothesis, as shown and discussed in the results section. We have shown how the various speeds affect the error by isolating it and then, for various collision angles, deriving optimal speed from the previous set of experiments. Our experiment was done in a very controlled environment, minimising the external factors' effect on the sensors as much as possible.

Future work on this going through can be done by taking off the controlled variable and moving slowly more and more towards emulating real-world case environmental conditions. This could help us improve the results by taking the output of the encoder-based kinematics model, making it more sturdy for future advanced experiments with the Pololu Robot.

References

[1] D. D. F. T. B. A. S.-N. R. T. A.-a. A.-m. Thomas Lew, Tomoki Emmei, "Contact inertial odometry: Collisions are your friends," *International Symposium on Robotics Research (ISRR)*, 2019.

[2] A. A.-S. Sami Haddadin, Alessandro De Luca, "Robot collisions: A survey on detection, isolation, and identification," *International Symposium on Robotics Research (ISRR)*, 2019.

[3] D. P. O'Dowd, "Bump sensor coursework lab sheet." https://colab.research.google.com/github/paulodowd/EMATM0054$_22-23/blob/main/Labsheets/Supp/SL2_BumpSensors.ipynb$.

[4] P. Robotics and Electronics, "line and bump sensors." https://www.pololu.com/docs/0J83/5.5.

[5] D. P. O'Dowd, "Odometry lab sheet." https://colab.research.google.com/github/paulodowd/EMATM0054$_22-23/blob/main/Labsheets/Core/L6_Odometry.ipynbscrollTo = mxkga8BEcRDQ$.

[6] P. I. Corke and P. Ridley, "Steering kinematics for a center-articulated mobile robot," *IEEE TRANSACTIONS ON ROBOTICS AND AUTOMATION*, vol. 17, no. 2, pp. 215–218, 2001.

[7] D. P. O'Dowd, "Pid lab sheet." https://colab.research.google.com/github/paulodowd/EMATM0054$_22-23/blob/main/Labsheets/Core/L7_PID.ipynbscrollTo = GkFszmZxscZM$.

[8] J. Borenstein and L. Feng, "Measurement and correction of systematic odometry errors in mobile robots," *IEEE TRANSACTIONS ON ROBOTICS AND AUTOMATION*, vol. 12, no. 6, pp. 869–880, 1996.

VI. Appendix

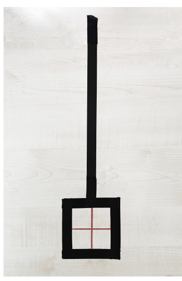


Fig. 11: Setup of the experiment